서울시 청년 1인 가구
임대 주택
청약 예측 서비스

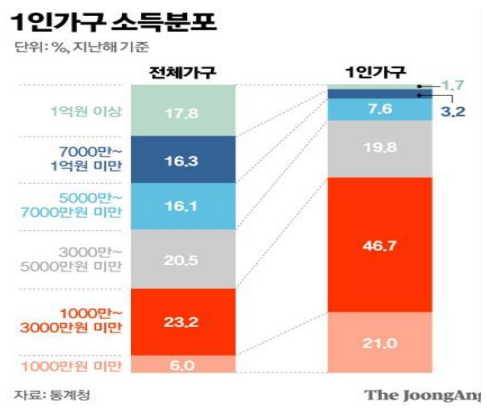
박 예 린

목차

1 기획 의도

20~30대 1인 가구, 시세보다 2~30% 저렴한 임대주택 인기

- 2030 청년 1인 가구 비율은 점점 늘어 가고 있으나
치솟는 집값으로 인한 주거비 부담과 전세사기 피해 확산으로
시세보다 저렴한 임대주택에 대한 관심이 늘고 있음



구분	보증금, 월세 등 주거비 보조	주택 관련 대출 및 이자 지원	공공임대주택 공급	자기 소유 지원	주택 개량 및 개보수 지원	정보 제공 및 상담
남성	65.8	67.1	67.0	67.2	62.3	63.1
20	70.9	71.1	71.0	71.8	64.7	65.6
30	66.9	69.0	67.7	68.6	63.6	64.8
40	64.3	66.9	68.8	66.6	62.0	63.8
50	64.2	64.6	65.0	66.3	61.0	62.1
여성	66.3	66.5	67.0	66.2	63.4	63.1
20	70.6	68.0	68.9	69.5	64.3	65.5
30	70.1	70.7	70.9	68.4	66.0	64.4
40	65.7	69.0	66.3	67.3	62.6	64.9
50	62.1	65.1	65.3	65.0	61.7	64.4

*출처 : 서울시 1인가구 실태조사 및 제도개선 연구용역 보고서

대한경제 | 2023.04.13.

[마피 사기주의보] 전세사기에 임대주택으로 몰리는 세입자들

빌라왕 **전세사기** 여파로 인해 2030세대 세입자들이 보증금 리스크가 낮은 **임대주택** 등을 주목하고 있다.... 합리적인 **임대**료로 주거 안정이 보장되는 민간**임대주택**이 내 집 마련 대안으로 급부상하면서 높은 **인기**를 끌고...

에너지경제 | 2022.10.06.

2030 영끌족은 옛말...LH 청년 매입**임대주택** '**인기**'

전세사기·깡통**전세**에 대한 불안 등도 영향을 준 것을 분석된다. 매입**임대**는 LH에서 주도하는 **주택** 제도로 안전하다는 인식이 있다. 특히 청년매입**임대주택**은 최장 6...

국제신문 PiCK | 10면 1단 | 2023.07.27. | 네이버뉴스

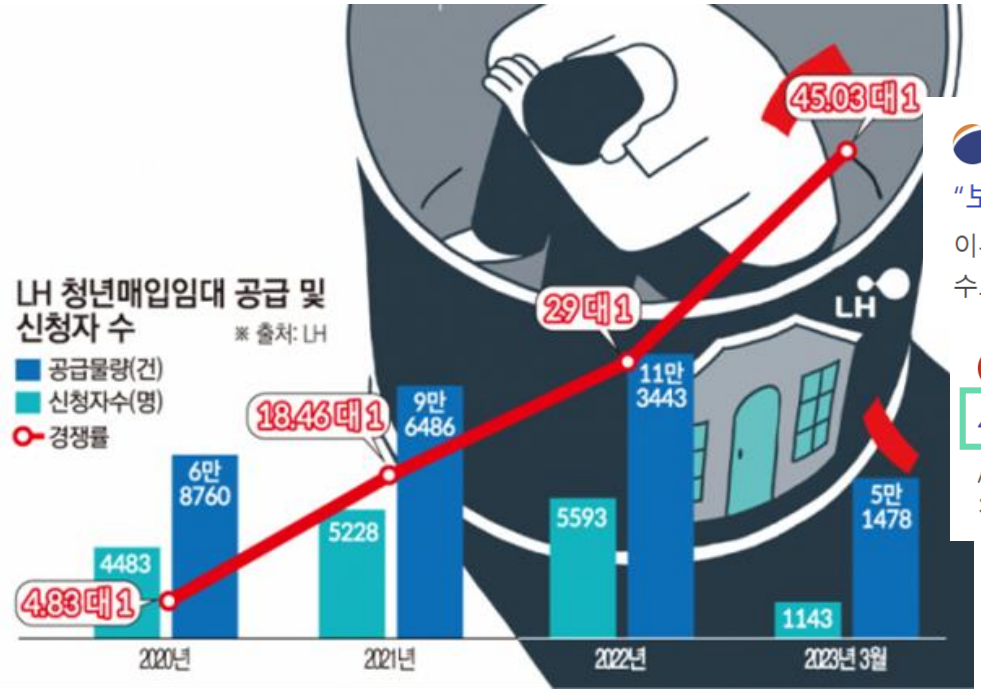
전세사기 걱정없는 LH행복**주택** **인기**

행복**주택**의 높아진 **인기**를 실감케 했다. 부산 모라는 100가구 모집에 246명 지원(경쟁률 2.5 대 1)했고, 울산 송정은 220가구 모집에 531명(2.4 대 1)이 몰렸다. LH...

1 기획 의도

공급 물량에 비해 엄청난 경쟁률

- 당해 공고의 매물량이 많지 않아 치열한 경쟁률
주택공사에서 따로 제공하는 청약 지표가 없어서
어느 지역, 어떤 단지에 청약을 넣어야 유리할지 판단이 어려운 상황



2022년 2차 청년 매입임대주택 입주자모집공고(2022.12.30.)
인터넷 청약신청 최종 경쟁률 공지

이투데이 | 2023.04.25.

"보증된 집에 살자"...올해 수도권 LH 청년매입**임대** 경쟁률 '45대 ...

이투데이=정용욱 기자 | **전세사기** 사태 확산 이후 공공 **임대주택** 선호 급증 올해 수도권 매입**임대 주택 인기**가 천정부지로 치솟고 있다. 청년 매입**임대주택** 경쟁...

시사저널이코노미 | 2023.04.16.

432가구 모집에 4만명 신청..공공**임대**에 몰리는 청년들

/ 사진=연합뉴스 수도권 공공**임대주택**이 청년들로부터 **인기**를 끌고 있다. 빌라왕 **전세사기** 여파로 보증금이 높은 **주택**을 꺼리는 현상이 심화된 데다 '**전세**의 월세...

						공급호수(호)	신청현황(누계) 신청자(명)	신청현황(누계) 경쟁률
재공급	강남구	백년빌	-	서울특별시 강남구 도곡로18길 10, 백년빌 (도곡동 547-11)	-	2	200	100
신규공급	강남구	희준빌	27B	서울특별시 강남구 논현로121길 35, 희준빌 (논현동 173-5)	-	8	631	78.9
신규공급	강남구	희준빌	31B	서울특별시 강남구 논현로121길 35, 희준빌 (논현동 173-5)	-	2	108	54
신규공급	강남구	희준빌	34A	서울특별시 강남구 논현로121길 35, 희준빌 (논현동 173-5)	-	2	420	210
						2	62	31
						2	22	11
						2	12	6
						2	9	4.5
						5	159	31.8
						1	115	115
						1	92	92
						1	174	174
						1	120	120
						1	131	131
						1	113	113

*출처 :sh주택공사

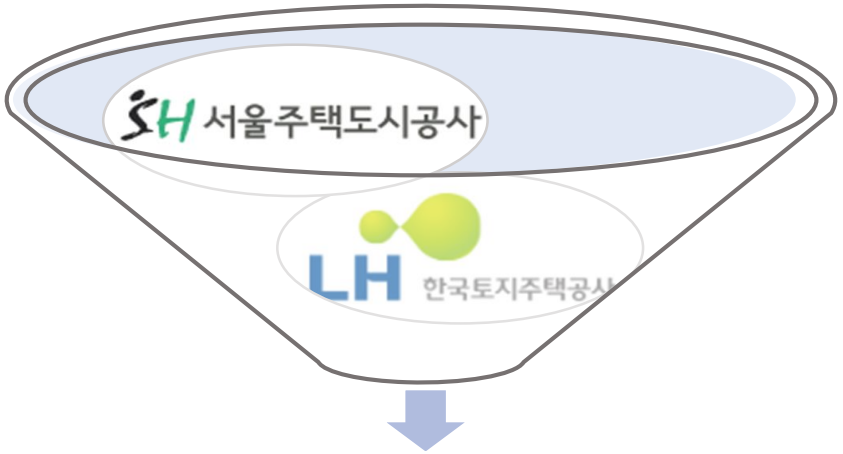
1 개발 목표

서울시 1인 가구 임대주택 청약 경쟁률 예측 서비스

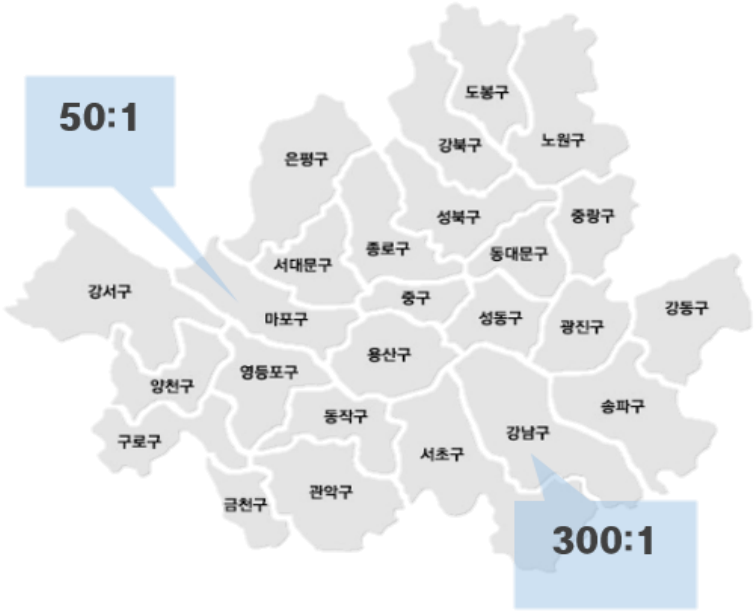
✓비싼 월세에 부담을 느끼는 1인가구 청년

✓치열한 임대주택 청약 경쟁률 속에서 청약 신청의 지표가 필요한 청년들

- 서울시 지역별 청년 임대주택 경쟁률 분석 후 예측하여 시각화



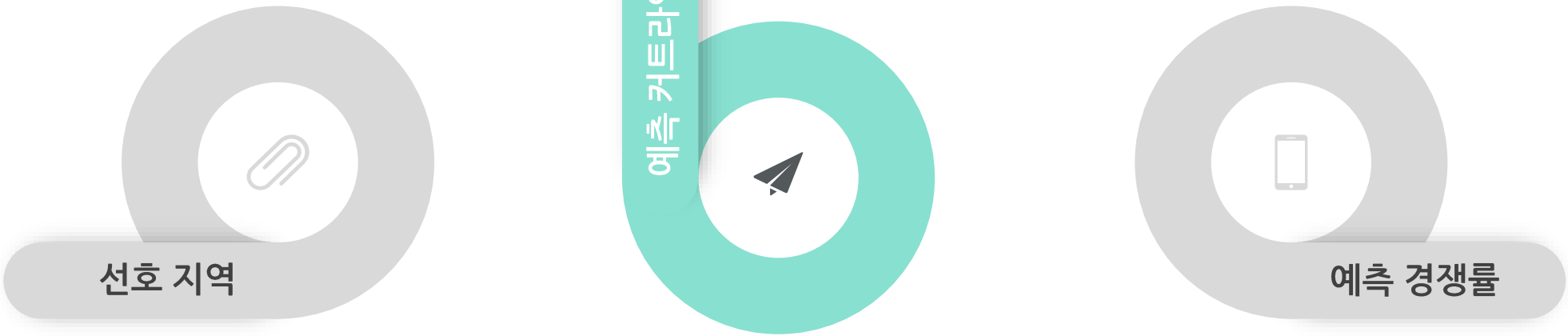
청년 1인 가구



1 개발 목표

지역별 청약 커트라인 예측 서비스

- 사용자의 순위, 가산점과 지역별 예측 커트라인을 비교하여 제공
당해 공고 선호 지역별 사용자 수를 실시간 제공
-청약 성공 확률을 높일 수 있음



현재 50명이 마포구를 선호하고 있습니다.
마포구 선호 비율은 30%입니다.

사용자님의 가산점보다
커트라인이 낮은 지역은
마포구, 구로구, 금천구 입니다.

마포구 예측 경쟁률은 50:1입니다.
마포구는 지역별 예측 경쟁률 2위 입니다.

2 사용 기술 목록표

Language	 1.8.0_371				
개발 환경	 4.3.9		 2019.09	 11.2.0.2	 4.3.1
Server	 9.0				

2 사용 기술 경험

- java1.8 기반 미니 프로젝트
- 은행 입출금 시스템
- 주소록 관리 시스템
- 도서 대여 프로그램
- single linkedlist 구현

- Spring 4.3.9 version
- 전자정부 프레임워크3.9 에서 Spring Framework를 이용해 CRUD 를 구현
- MVC2패턴 게시판, 회원관리 프로그램

- 프로젝트 UI 제작
- bootstrap 반응형 페이지 제작

- 회원관리 유효성 검사
- 비동기 통신(AJAX)을 구현
- chart.js로 그래프 구현

- MVC2방식의 임대주택 청약 예측 서비스 웹사이트 구현
- 회원관리 시스템(등록, 삭제, 수정)
- 게시판 (등록, 삭제, 페이징 처리)

- 데이터 베이스 설계 (ERD)
- JOIN, Group by 등과 같은 쿼리문 작성
- 게시판 / 회원관리 등의 테이블 관리

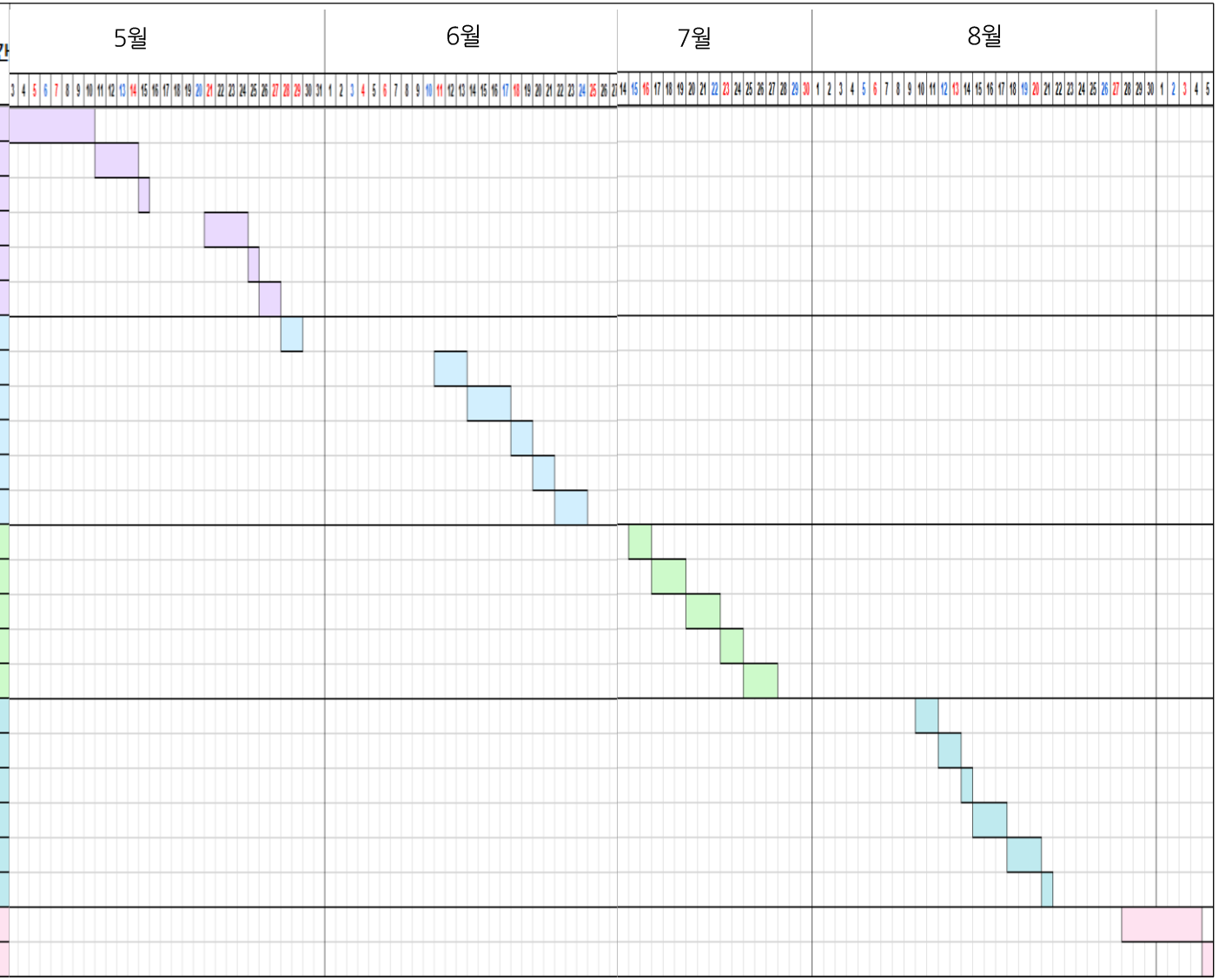
- 데이터 분석
- 회귀 모델로 예측
- 공공데이터 api와 연동

- 기본적인 문법 학습

- Ubuntu 기반 Linux개발 환경 구축 실습
- 리눅스 커맨드 실습

3 개발 스케줄표

업무 분장	번호	내용	시작일	종료일	작업기간
기획	1	아이디어 구상	2023-05-01	2023-05-10	10
	2	기획안 작성	2023-05-11	2023-05-14	4
	3	기획안 중간 발표	2023-05-15	2023-05-15	1
	4	개발 스케줄표 작성	2023-05-27	2023-05-28	4
	5	기획안 최종 발표	2023-05-25	2023-05-25	1
	6	요구사항 정의서 작성	2023-05-26	2023-05-27	2
UI 개발	9	화면 설계서 작성	2023-05-28	2023-05-29	2
	10	예측 경쟁률 지역별 지도위에 표시 UI 구현	2023-06-11	2023-06-13	3
	11	경쟁률 예측 서비스 메인페이지 UI 구현	2023-06-14	2023-06-17	4
	12	내 가산점 계산하기 페이지 UI구현	2023-06-18	2023-06-19	2
	13	청약 예측 dashboard UI 구현	2023-06-20	2023-06-21	2
	14	UI 전체 수정	2023-06-22	2023-06-24	3
청약 경쟁률 데이터 분석	15	청약 경쟁률,커트라인 데이터 수집	2023-07-15	2023-07-16	2
	16	데이터 전처리작업	2023-07-17	2023-07-19	3
	17	청약 경쟁률 데이터 분석 및 예측	2023-07-20	2023-07-22	3
	18	청약 커트라인 데이터 분석 및 예측	2023-07-23	2023-07-24	2
	19	데이터 결과 보고서 작성	2023-07-25	2023-07-27	3
Servlet&JSP개발	20	UML 작성, ERD 설계	2023-08-10	2023-08-11	2
	21	예측 경쟁률 지역별 조회 기능 구현	2023-08-12	2023-08-13	2
	22	내 가산점 계산하기 기능 구현	2023-08-14	2023-08-14	1
	23	사용자의 선호지역에 대한 정보 출력기능 구현	2023-08-15	2023-08-17	3
	24	사용자의 가점보다 커트라인이 낮은 지역 출력 기능 구현	2023-08-18	2023-08-20	3
	25	중간 점검 및 발표	2023-08-21	2023-08-21	1
Spring 개발	26	Spring framework 개발	2023-08-28	2023-09-04	8
	27	발표	2023-09-05	2023-09-05	1



4 빅데이터 분석 결과보고서

서울시 1인 가구 임대주택 지역별 청약 커트라인, 경쟁률 예측 데이터 분석

수집 → 전처리 및 검증 → 분석 + 예측(모델링) → db

SH 서울주택도시공사

행정안전부

N Pay 부동산

4 빅데이터 분석 결과보고서

지역별 경쟁률, 커트라인 예측 데이터 수집

✓SH 주택 공사 제공하는 기본 데이터

- **rate** : 경쟁률
- **size** : 평형
- **supply** : 공급 세대수
- **value** : 임대주택 월세 (보증금 3천만원 기준)
- **people** : 매물별 지원자 수
- **cut** : 청약 커트라인

서울시 1인가구 실태조사에 따르면 청년 1인가구 밀집지역은
학교, 직장근처 번화가의 접근성, 편의시설에 영향을 많이 받음.

✓네이버 부동산 제공

- **d.sub** : 지하철 역까지의 거리,
- **d.hos** : 병원까지의 거리

지역별 특성을 반영하기 위해

✓행안부 제공

- **security** : 지역별 사회안전지수(경제활동, 생활안전, 건강보건, 주거환경의 지표 반영),
- **avg.price** : 지역별 평균 집값

✓**경쟁률** 예측 독립변수

x1 : size
x2 : supply
x3 : value
x4 : d.sub
x5 : d.hos
x6 : security
x7 : avg.price
x8 : people

✓경쟁률 예측 종속변수

y : rate

✓**커트라인** 예측 독립변수

x1 ~ x7 동일
x8 : rate

✓커트라인 예측 종속변수

y : cut

4 빅데이터 분석 결과보고서

지역별 경쟁률 예측 모델링 및 결과 분석

```
> model1 <- lm(rate ~ size + supply + value + d.sub + d.hos + security
+              + avg.price + people , data = testset)
> summary(model1)

Call:
lm(formula = rate ~ size + supply + value + d.sub + d.hos + security +
    avg.price + people, data = testset)

Residuals:
    Min      1Q  Median      3Q     Max
-346.82  -38.55  -15.46   12.11  714.70

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.829e+02  8.601e+01   3.289  0.00103 **
size         2.896e+00  4.045e-01   7.160 1.41e-12 ***
supply      -9.933e-01  1.140e-01  -8.713  < 2e-16 ***
value       -3.177e-04  4.035e-05  -7.873 7.77e-15 ***
d.sub       -3.897e-02  9.591e-03  -4.064 5.15e-05 ***
d.hos       -8.663e-03  1.138e-02  -0.761  0.44681
security    -5.817e+00  1.701e+00  -3.419  0.00065 ***
avg.price    2.063e-02  3.190e-03   6.467 1.46e-10 ***
people       1.892e-01  7.282e-03  25.988  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 91.67 on 1185 degrees of freedom
Multiple R-squared:  0.4211,    Adjusted R-squared:  0.4172
F-statistic: 107.8 on 8 and 1185 DF,  p-value: < 2.2e-16
```

✓다중선형 회귀 모델

- 여러개의 독립변수를 다루는 다중선형 회귀 모델 사용

✓결측치 처리

- 평균값 대체

✓회귀식

- rate = 282.9 + 2.896×size − 0.9933× supply− 0.0003177× value − 0.03897×d.sub − 0.008663×d.hos − 5.817×security + 0.02063×avg.price + 0.1892×people

✓p-value

- 2.2e-16 으로 신뢰수준이 95%이상임을 나타냄.

✓Adjust R-squared

- 모델의 설명력을 나타내며 0.4172 로 다소 낮은 값.

현실을 41%정도 설명할 수 있음.

4 빅데이터 분석 결과보고서

지역별 커트라인 예측 데이터 전처리 및 모델링

```
# 결측치 처리 (중앙값으로 대체)
data$size <- ifelse(is.na(data$size), median(data$size, na.rm = TRUE), data$size)
data$people <- ifelse(is.na(data$people), median(data$people, na.rm = TRUE), data$people)
data$rate <- ifelse(is.na(data$rate), median(data$rate, na.rm = TRUE), data$rate)
data$value <- ifelse(is.na(data$value), median(data$value, na.rm = TRUE), data$value)
data$d.sub <- ifelse(is.na(data$d.sub), median(data$d.sub, na.rm = TRUE), data$d.sub)
data$security <- ifelse(is.na(data$security), median(data$security, na.rm = TRUE),
data$security)
data$avg.price <- ifelse(is.na(data$avg.price), median(data$avg.price, na.rm = TRUE),
data$avg.price)
data$supply <- ifelse(is.na(data$supply), median(data$supply, na.rm = TRUE), data$supply)
data$cut <- ifelse(is.na(data$cut), median(data$cut, na.rm = TRUE), data$cut)
data <- na.omit(data)

for (col in names(trainData)) {
  trainData[[col]][is.na(trainData[[col]])] <- mean(trainData[[col]], na.rm = TRUE)
}

for (col in names(testData)) {
  testData[[col]][is.na(testData[[col]])] <- mean(testData[[col]], na.rm = TRUE)
}



# 데이터 분할
set.seed(42)
trainIndex <- sample(1:nrow(data), 0.8*nrow(data))
trainData <- data[trainIndex,]
testData <- data[-trainIndex,]

# 랜덤포레스트 모델 학습
rf_model <- randomForest(cut ~  d.hos + size  + rate + value + d.sub + security + avg
.price + supply, data=trainData, ntree=100)
```

✓랜덤 포레스트

- 지역별 청약 커트라인은 연속성 데이터가 아니기 때문에 랜덤포레스트를 선택
- 수백 개의 의사결정나무를 결합하여 이를 평균화한 모형을 생성하는 앙상블 학습 방법

✓전처리

- 데이터는 80%의 훈련 데이터와 20%의 테스트 데이터로 분할
- 결측치 처리 (중앙값으로 대체) 및 제거
- 훈련 데이터의 결측치 처리 (평균값으로 대체)
- 테스트 데이터의 결측치 처리 (평균값으로 대체)

✓100개의 트리를 사용하여 모델을 학습

4 빅데이터 분석 결과보고서

지역별 커트라인 예측 결과 분석 및 한계

```
# 예측
predictions <- predict(rf_model, newdata=testData)

results <- data.frame(region = testData$region, Predicted = predictions)

# 지역별 예측값의 평균
region_avg_predictions <- aggregate(Predicted ~ region, data = results, FUN = mean)
```

```
> # 결과 출력
> print(region_avg_predictions)
      region Predicted
1     강남구  25.30619
2     강동구  20.83698
3     강북구  19.97523
4     강서구  24.31421
5     관악구  22.24189
6     광진구  25.55644
7     구로구  15.38101
8     금천구  17.34525
9     노원구  24.82833
10    도봉구  20.80770
11  동대문구  24.71734
12    동작구  22.70767
13    마포구  25.76011
14  서대문구  22.30989
15    서초구  25.74415
16    성동구  22.50904
17    성북구  22.45666
18    송파구  22.90332
19    양천구  17.24008
20  영등포구  22.28743
21    용산구  26.41230
22    은평구  17.70094
23    종로구  27.05767
24      중구  25.39868
25    중랑구  20.65259
```

```
> # 성능 평가 (RMSE)
> rmse <- sqrt(mean((predictions - testData$cut)^2))
> print(rmse)
[1] 1.679401
> mae <- mean(abs(predictions - testData$cut))
> # 결과 출력
> print(paste("MAE:", mae))
[1] "MAE: 1.13411872009695"
```

```
> # 특성 중요도 확인
> importance(rf_model)
             IncNodePurity
d.hos             3735.009
size              2865.574
rate             15395.639
value             3277.473
d.sub             4676.697
security          1987.411
avg.price         2428.313
supply            2181.874
```

✓성능 평가

- RMSE는 예측 오차의 제곱합의 평균의 제곱근으로,
값이 작을수록 예측의 정확도가 높다는 것을 의미.
- MAE (Mean Absolute Error): MAE는 실제 값과 예측 값의 차이를
절댓값으로 변환하여 평균한 것으로,
값이 작을수록 예측의 정확도가 높다는 것을 의미.

✓예측 변수의 중요도 지수

- 모델의 예측 변수별 중요도를 확인한 결과,
경쟁률 > 지하철역까지의 거리 >병원까지의 거리

✓한계

- 데이터셋의 수가 1185개로 부족함
- 데이터가 있는 예측변수의 수가 부족함

청약가점 및 경쟁률에 영향을 미치는 다양한 외부 요인들이 더 있을 수 있으며, 완전한 결과를 도출하는 데 한계가 있음.

5 요구사항 정의서

3. 기능적 요구사항

3.1 청약 진단하기 기능

3.1.1 아이디 입력

- 사용자는 자신의 아이디를 텍스트 형식으로 입력할 수 있어야 한다.
- "아이디를 입력하세요" 라는 안내 메시지와 함께 입력 폼 제공

3.1.2 선호 지역 선택

- 사용자는 드롭다운 메뉴에서 선호 지역을 선택할 수 있어야 한다.
- 마포구, 강남구, 강서구 등의 서울시 자치구 옵션 제공

3.1.3 순위 선택

- 사용자는 자신의 순위(1 순위, 2 순위, 3 순위) 중 하나를 버튼을 통해 선택할 수 있어야 한다.
- 각 순위에 대한 상세 설명 제공

3.1.4 가산점 선택

- 사용자는 자신에게 해당되는 가산점 항목을 체크박스를 통해 선택할 수 있어야 한다.
- 각 가점에 대한 설명 제공하며, 복수 선택 가능

3.1.5 결과 제출

- 사용자는 입력된 정보를 바탕으로 가점을 계산하고 결과값을 얻기 위해 폼을 제출할 수 있어야 한다.

3.2 청약 예측 및 분석 기능

- 사용자가 입력한 순위와 가점을 바탕으로 청약 예측정보를 제공할 수 있어야 한다.

3.2.1 선호지역 카드

- 사용자의 선호지역을 표시한다.

3.2.2 선호지역 예측 경쟁률 카드

- 사용자가 입력한 선호지역에 대한 예측 경쟁률을 보여줄 수 있어야 한다.

3.2.3 선호지역 예측 커트라인 카드

- 사용자가 입력한 선호지역에 대한 예측 커트라인을 보여줄 수 있어야 한다.
- 커트라인 점수에 따른 프로그레스 바를 통해 시각적으로 표시한다.

3.2.4 월별 경쟁률 차트

- 사용자의 선호지역의 월별 실제 경쟁률을 차트로 시각화하여 표시한다.

3.2.5 지역별 경쟁률 순위 파이 차트

- 각 지역의 경쟁률 상위 5 개 지역을 파이 차트로 시각화하여 표시한다.

3.2.6 사용자의 가점보다 예측 커트라인이 낮은 지역 리스트

- 사용자의 순위와 가산점보다 예측 커트라인이 낮은 지역과 해당지역의 커트라인을 리스트 형태로 모두 출력하여 표시한다.
- 각 지역들의 예측 커트라인을 프로그레스 바를 통해 시각적으로 표시한다.

3.2.7 페이지 이동

- "처음으로" 버튼을 통해 가산점 계산하기 페이지로 이동할 수 있어야한다.

3.3 메인페이지

3.3.1 네비게이션 바

- 홈, 모집공고, 청약진단하기, 상세조회, ABOUT 등의 메뉴를 포함한다.
- 청약 진단하기 및 상세조회 메뉴는 드롭다운 형태로 하위 메뉴를 포함한다.
- 검색 모달이 있으며 버튼 클릭 시 사용자가 검색어를 입력할 수 있다.

3.3.2 캐러셀 기능

- 캐러셀 영역에 이미지 전환과 함께 멘트를 표시할수 있어야 한다.

3.3.3 지도 조회 기능

- 지도상의 자치구 영역에 마우스 호버 시 예측 경쟁률 박스로 표시할 수 있어야 한다.

3.1.4 매물 조회 기능

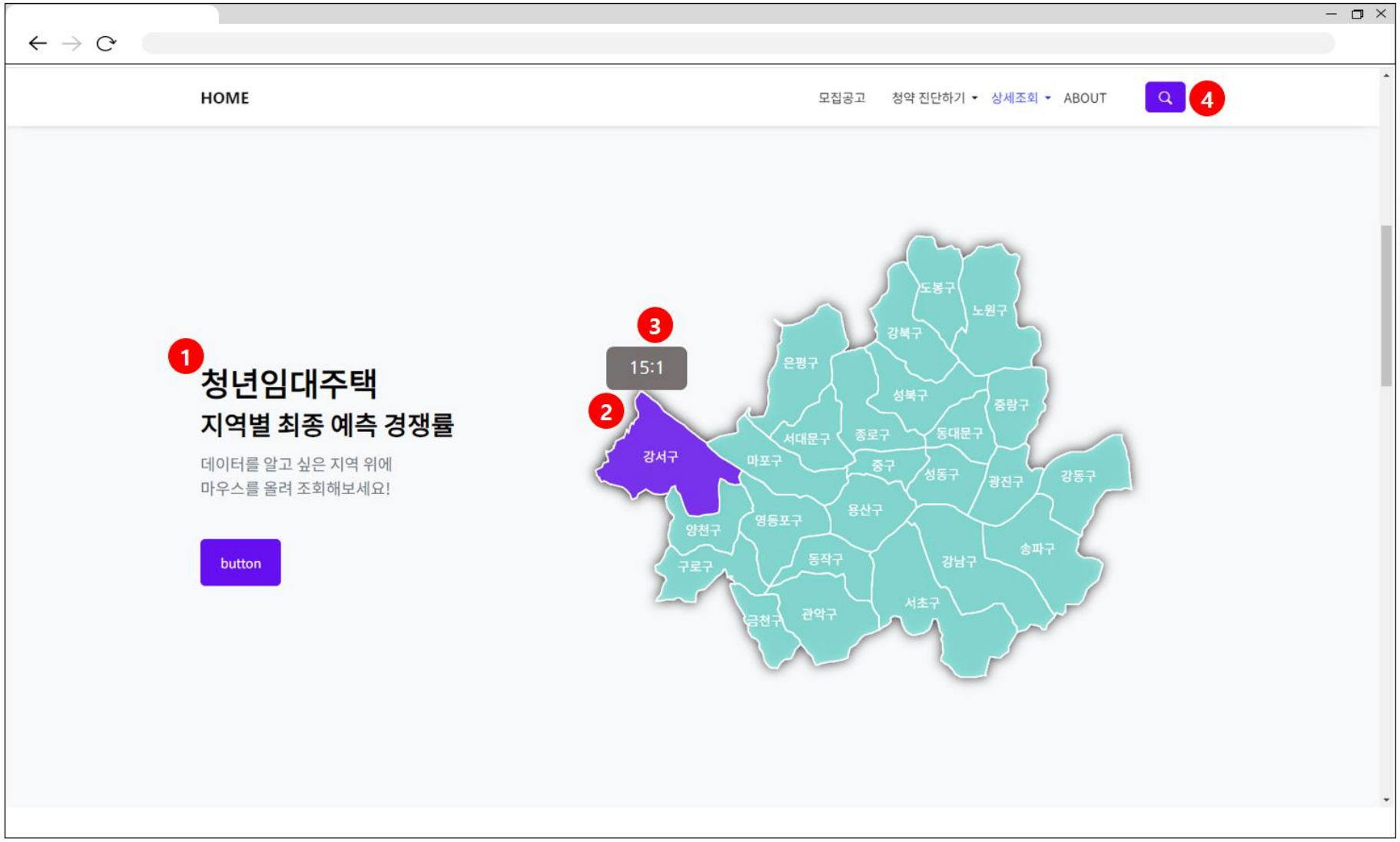
- 사용자가 지역을 선택하면 해당지역의 경쟁률이 높았던 단지 5 개를 카드형태의 캐러셀로 표시해 보여줄수 있어야한다.

3.1.5 서비스 이동 기능

- 각 기능을 클릭하면, 해당 기능과 관련된 페이지나 섹션으로 이동 할수 있어야 한다.

6 화면 설계서

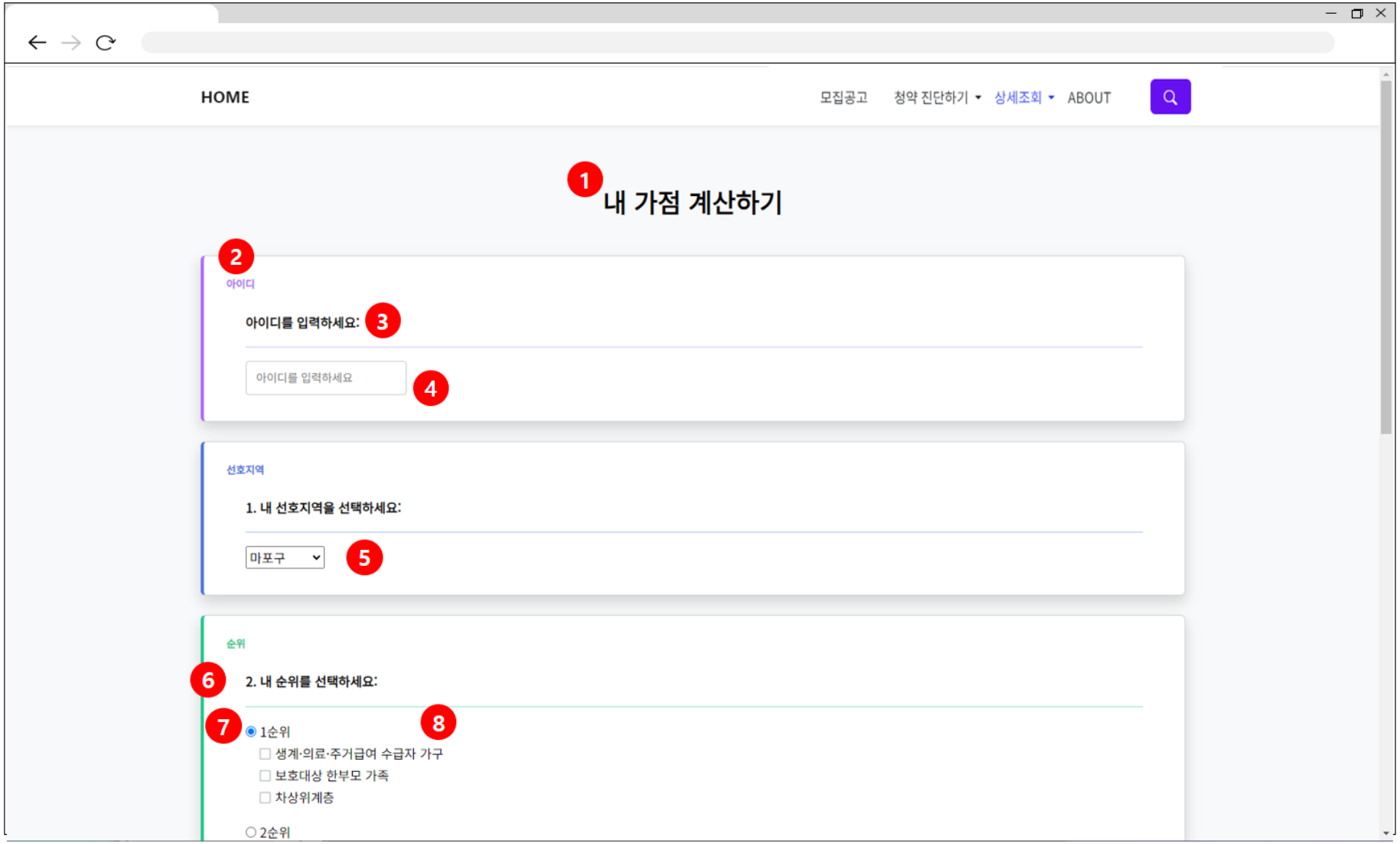
화면 이름	경쟁률 예측 페이지	화면 경로	메인 페이지 > 경쟁률 예측 페이지	화면 ID	-	작성일	2023/05/30



Description	
1	제목 표시
2	지도에서 자치구 영역에 마우스 오버 시 최종 예측 경쟁률 띄워 표시
3	예측 경쟁률 표시 플로팅 박스
4	돋보기를 클릭하면 검색 모달창

6 화면 설계서

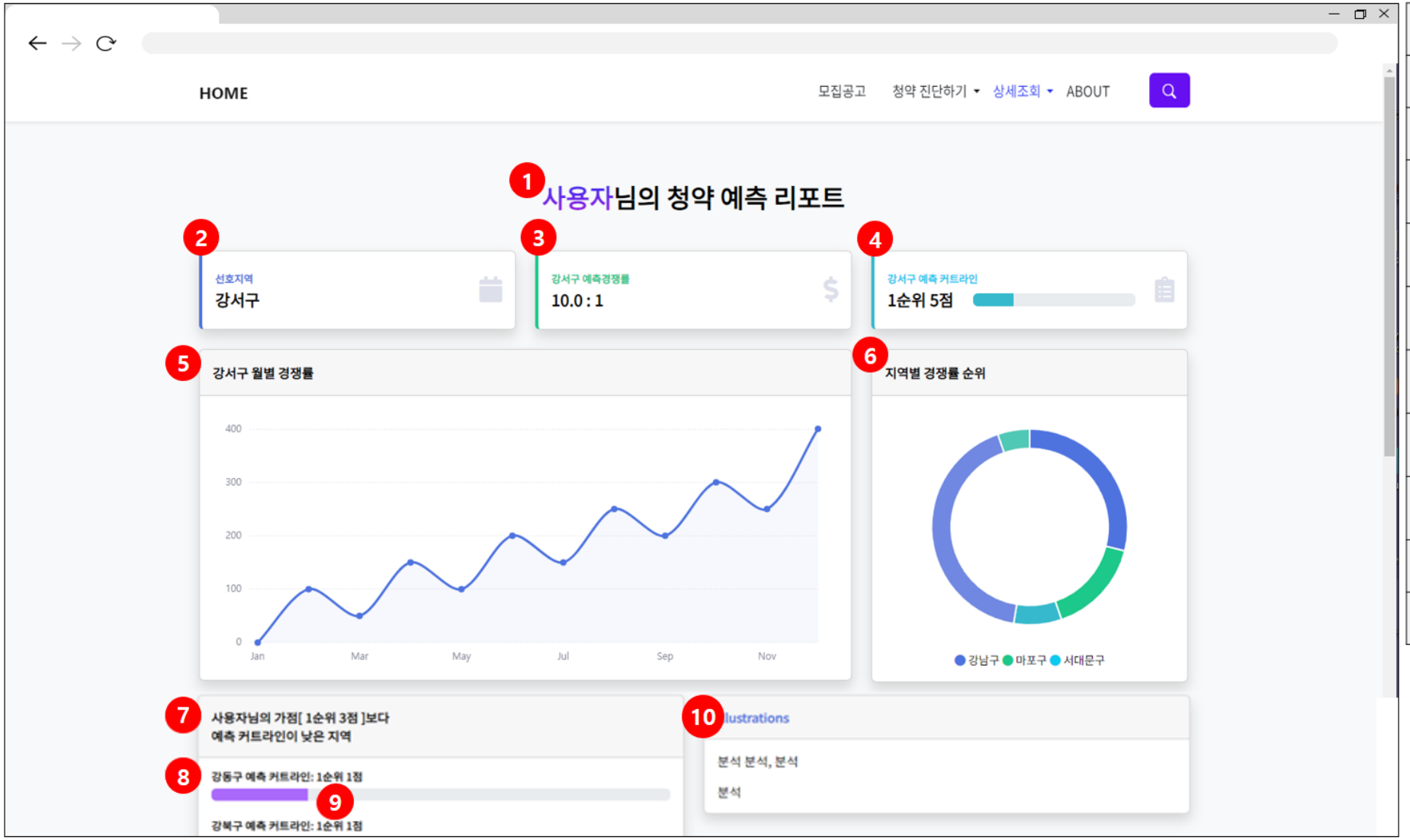
화면 이름	청약 진단 페이지	화면 경로	청약 진단하기 > 내 가점 계산하기	화면 ID	-	작성일	2023/05/30



Description	
1	제목 표시
2	카드 소제목 표시
3	설문 메시지
4	아이디 입력 폼
5	지역 선택 드롭 다운
6	설문 항목에 대한 제목
7	순위 선택 라디오 버튼
8	각 순위에 대한 설명 제공

6 화면 설계서

화면 이름	청약 예측 분석 페이지	화면 경로	청약 진단하기 > 청약 예측 분석 페이지	화면 ID	-	작성일	2023/05/30

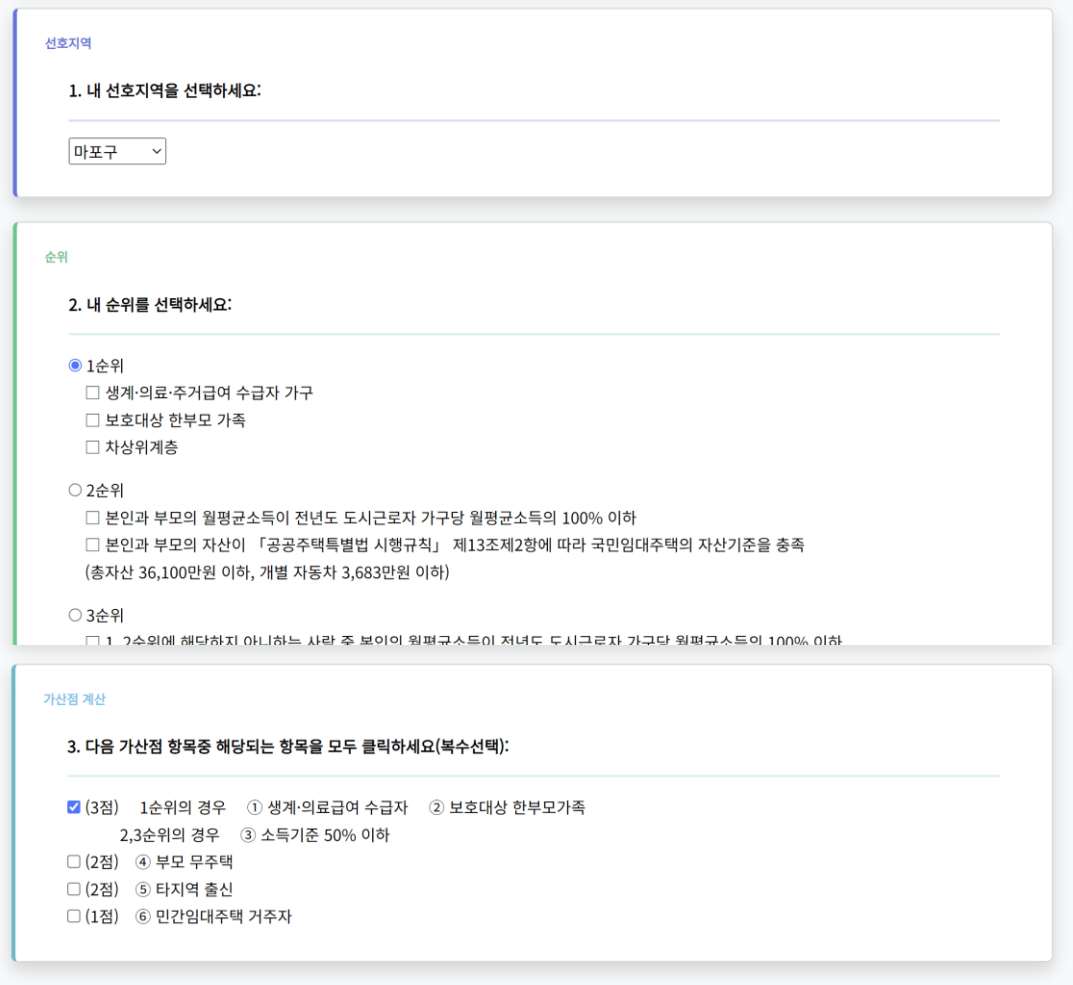


Description	
1	제목 표시
2	사용자의 선호지역 표시
3	사용자의 선호 지역의 예측 경쟁률 표시
4	사용자의 선호 지역의 예측 커트라인 표시
5	선호 지역의 월별 경쟁률 꺾은선 그래프
6	경쟁률 상위 5개 지역 파이 그래프
7	사용자의 가점보다 예측커트 라인이 낮은 지역 표시
8	리스트 형태로 모두 표시, 지 역별 커트라인 함께 출력
9	프로그레스 바로 시각화
10	추후 분석 결과 제공

7 주요 서비스 기능

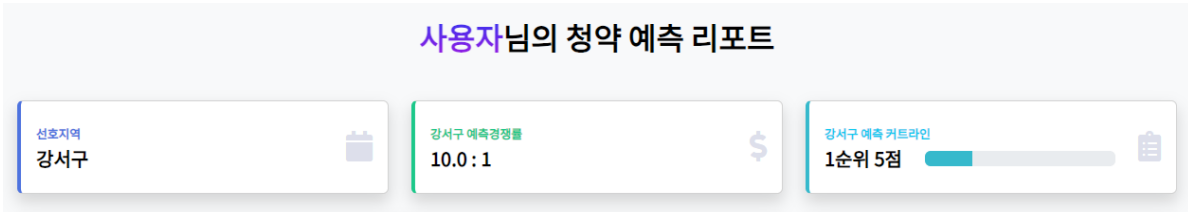
가산점 계산하기

✓순위와 가산점은 현재 소득과 자산에 따라 달라지므로
각 항목에 대한 설명 제공해 현재 상황 파악 가능



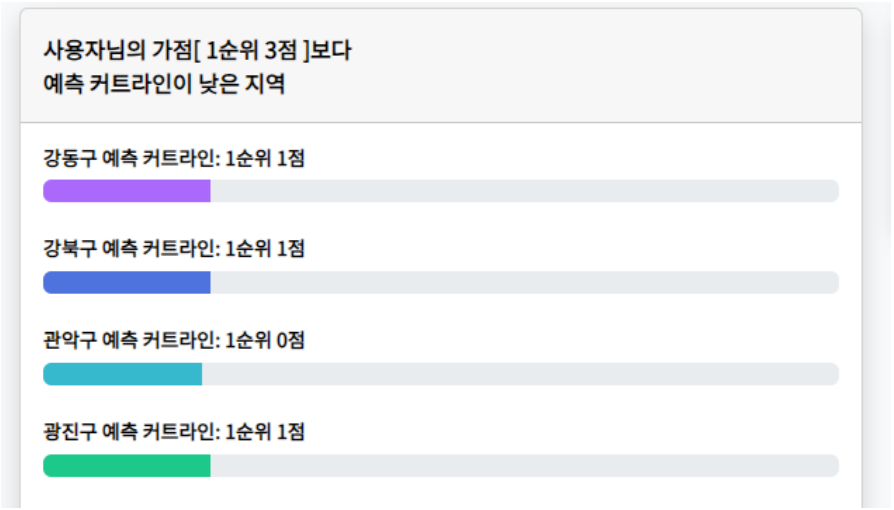
사용자의 선호지역에 대한 경쟁률과 커트라인 정보 출력

✓내가 살고 싶은 동네에 대한 궁금증 해결

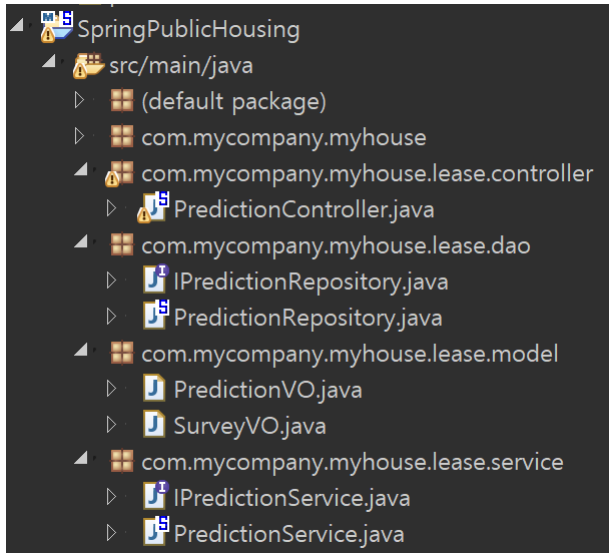


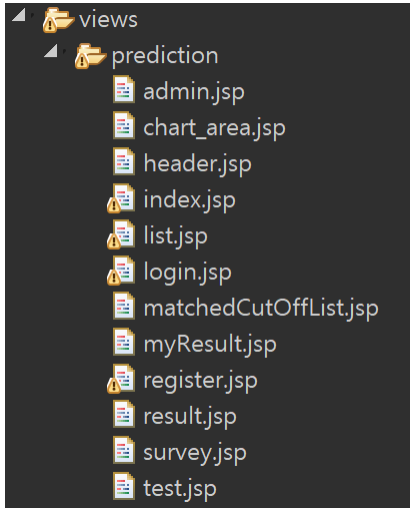
사용자의 가점보다 예측 커트라인이 낮은 지역 출력

✓당해 공고에 어느 지역에 청약을 넣으면 좋을지 지표 마련



8 주요 소스 코드





Package Explorer

```java
@Controller
public class PredictionController {

    @Autowired
    private PredictionService predictionService;

    @RequestMapping(value="/insert")
    public String insertPrediction(
            @RequestParam("district") String district,
            @RequestParam("competitionRate") double competitionRate,
            @RequestParam("cutOff") int cutOff,
            RedirectAttributes redirectAttributes) {

        PredictionVO prediction = new PredictionVO(district, competitionRate, cutOff);

        try {
            predictionService.insertPredictions(prediction);
            redirectAttributes.addFlashAttribute("message", "Prediction has been added.");
        } catch (Exception e) {
            redirectAttributes.addFlashAttribute("message", e.getMessage());
        }

        return "redirect:/predictions";
    }

    @RequestMapping(value="/preferred")
    public String getPreferredPrediction(@RequestParam("preferredDistrict") String preferredDistrict, Model model) {
        PredictionVO prediction = predictionService.getPrediction(preferredDistrict);
        model.addAttribute("preDto", prediction);

        return "predictions/preferredView";
    }

    @RequestMapping(value="/lower")
    public String getLowerCutoffDistricts(HttpSession session, Model model) {
        SurveyVO survey = (SurveyVO) session.getAttribute("sdto");
```

Controller

✓ url mapping과정에서 익숙치 않아 어려움을 겪음

8 주요 소스 코드

```java
@Repository
public class PredictionRepository implements IPredictionRepository {

    @Autowired
    private JdbcTemplate jdbcTemplate;


      @Override
      public List<PredictionVO> plist() {
          String query = "select district, competition_rate, cut_off from predictions";
          return jdbcTemplate.query(query, new PredictionVOMapper());
      }

      @Override
      public PredictionVO getPrediction(String preferredDistrict) {
          String query = "SELECT p.district, p.competition_rate, p.cut_off FROM predictions p WHERE p.district = ?";
          return jdbcTemplate.queryForObject(query, new Object[]{preferredDistrict}, new PredictionVOMapper());
      }

      @Override
      public List<PredictionVO> getLowerCutoff(int extrapoints) {
          String query = "SELECT district, cut_off FROM Predictions WHERE cut_off < ?";
          return jdbcTemplate.query(query, new Object[]{extrapoints}, new PredictionVOMapper());
      }


}
```

Repository

✓ 서비스의 주된 기능이 비교 출력해 주는 기능이여서 query 문에 신경을 썼으나 더 간단한 mybatis로 전환하지 못해 아쉬움

8 주요 소스 코드

```java
package com.mycompany.myhouse.lease.service;


import java.util.List;

public interface IPredictionService {
    void insertSurvey(String userId, String preferredDistrict, int rank, String[] extraPointsItems);
    void insertPredictions(PredictionVO pred);
    public List<SurveyVO> slist();
    public List<PredictionVO> plist();
    public PredictionVO getPrediction(String preferredDistrict);
    public List<PredictionVO> getLowerCutoff(int extrapoints);


}
```

Service

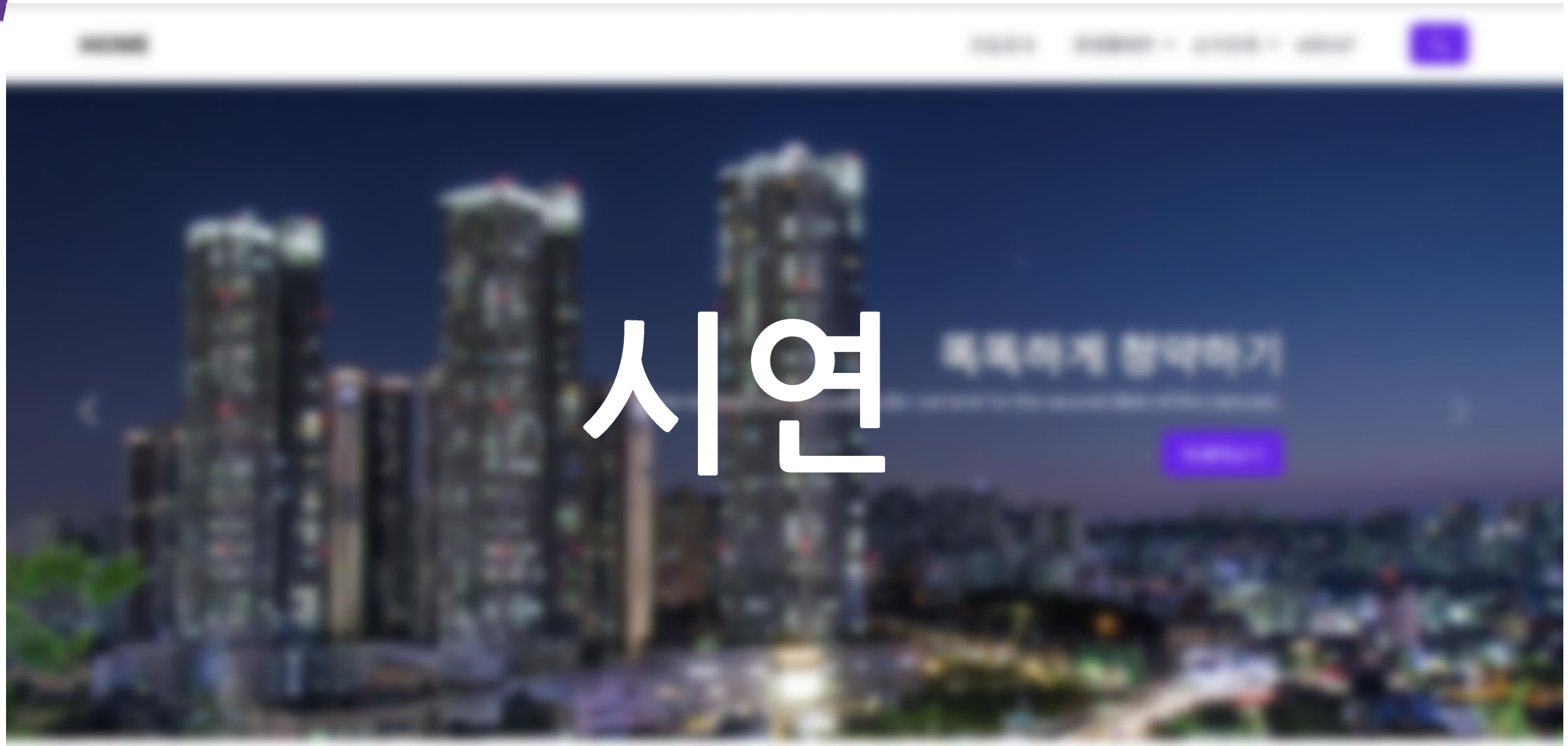
✓유연성을 높이기 위해 인터페이스 구현

```java
int rankPoints = (3 - rank) * 10;
  int extraPoints = rankPoints;

  for (String item : extraPointsItems) {
      switch (item) {
          case "a":
              extraPoints += 3;
              break;
          case "b":
          case "c":
              extraPoints += 2;
              break;
          case "d":
              extraPoints += 1;
              break;
      }
  }
```

```java
function calculate(points) {
    var tens = Math.floor(points / 10);
    var ones = points % 10;

    switch (tens) {
        case 2: return "1순위 " + ones;
        case 1: return "2순위 " + ones;
        case 0: return "3순위 " + ones;
        default: return points + "점";
    }
}
```

사용자의 순위+가산점과 예측 커트라인 비교

✓순위에 10 단위 점수 부여 후 가산점과 합산하고
예측 커트라인과 비교 후, 다시 js로 순위와 가점 형태로 변환해 출력


시연

10 향후 계획 및 프로젝트 수행 소감

✓순위+가점과 예측 커트라인의 비교 로직

→ 더 간단한 알고리즘 고민

✓사이트의 접근성을 높이기 위해 비회원 기능 → 세션 유지, 중복 데이터의 처리

✓예측 모델의 정확도와 신뢰성을 더 높일 방법을 고려

✓초기 설계에서 많은 변형

✓시간에 쫓겨서 하나에 집중하지 못한 아쉬움

✓에러 핸들링

✓개발을 완료하지 못한 추가적인 기능들 완성

✓AWS를 통한 배포와 실제 런칭